Symphony SoundBite: Quick Start with Symphony Studio

Installation and Configuration

Document Number: DSPB56371UGQS
Rev. 2

September 2008

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 921 03 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-521-6274 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Contents

About This Document

This document details the installation and configuration particulars for Symphony Studio when used with the Symphony SoundBite development board.

Audience

This guide is intended for new users to Symphony Studio and the Symphony SoundBite development board.

Organization

This document is organized into the following chapters:

Chapter 1	Introduction to Symphony SoundBite
Chapter 2	Installation and initial configuration guide for Symphony Studio

References

For more information, see the following documents:

- ***Symphony SoundBite: Reference Manual***
- ***Symphony Studio: Eclipse for Symphony DSPs User's Guide***
- ***Symphony SoundBite: Assembly Project Template User's Guide***
- ***Symphony SoundBite: Demo Application User's Guide***

Chapter 1 Introduction

Symphony Studio is a new integrated development environment (IDE) that supports Freescale's Symphony family of 24-bit digital signal processors. Symphony Studio differs substantially from the previous suite of development tools for these DSPs. This document provides a brief introduction to this new IDE particularly as it relates to software development for the Symphony SoundBite target hardware.

This document is meant to supplement the user's guides included with the Symphony Studio installation. It contains information and configuration particular to the use of Symphony Studio with the Symphony SoundBite development board.

1.1 Symphony SoundBite Hardware

The Symphony SoundBite is an inexpensive but feature-packed solution for audio digital signal processing based on the DSPB56371 digital signal processor. The Symphony SoundBite supports the simultaneous input and output of eight channels of analog audio or six channels of analog with digital stereo optical SP/DIF through the eight 3.5 mm stereo jacks on the board. Sample rates of up to 192 kHz are supported with the four 24-bit stereo codecs mounted on the board. A serial I2C EEPROM provides non-volatile storage for the DSP as well as also allowing the board to operate stand-alone (DSP application code is stored in the EEPROM).

An eight-position DIP switch and nine LEDs are connected to GPIO pins on the DSP and are available for user interaction and indication. An expansion header is provided to facilitate off-board expansion.

The on-board USB interface is built around the FT-2232. It provides for low-level JTAG and OnCE debugging capability with the use of Symphony Studio. Symphony Studio allows application development in assembly alone or C and assembly. Additionally, it provides for high-level SPI and I2C communication via the DSP's SHI port, which provides data pipe between the host PC and the DSP. Data or application code can be transmitted through the data pipe.

Chapter 2 Configuring Symphony Studio

2.1 Installation

This installation guide refers to the production release of Symphony Studio 1.1.0. Newer versions might be different than what is described here. If you are installing a different version, refer to the notes for that release for more details.

Begin the installation process by launching the installation utility ***Symphony_Studio_IDE.exe***. Follow the instructions of the installer as they are presented. Pay particular attention to the installation path of Symphony Studio. No spaces are permitted in the pathname with the current release, so the usual “Program File” installation directory is not usable. The default C:\Symphony-Studio is preferred and is assumed throughout.

When the installation is complete, a dialog box with several checkbox options is presented. The built-in USB-to-JTAG command converter on the Symphony SoundBite board requires the FTD2XX drivers to be installed. Leave the FTD2XX box checked in this final dialog box and click ***Finish***.

2.2 Initial Configuration

Launch Symphony Studio. On the first launch, the Workspace Launcher dialog box will be presented. The default workspace name is “workspace” and is placed in the user’s working directory. The name and location may be changed. The default name and location is assumed throughout. Click ***OK***.

Once it is fully launched, the newly created workspace opens with a Welcome window.

Perform the following steps:

1. Click the ***Workbench*** icon, which looks like an arrow rolled over on itself.
2. Right-click the ***Resource*** button in the upper right corner of the window and select ***Close***.
3. Click the ***Open Perspective*** button near the top center and select ***Debug***.
4. Click the ***Open Perspective*** button again (now next to the ***Debug*** button on the upper right) and select ***C/C++***.

The two clickable buttons in the upper right corner of the Symphony Studio window are used for changing between the two perspectives, ***C/C++*** and ***Debug***. The ***C/C++*** perspective is where all the project management and editing or coding occurs. The ***Debug*** perspective is where the debugging activities occur.

The workspace is now configured in the default state that is assumed throughout.

2.3 External Tool

An external tool is the mechanism by which the Symphony Studio GUI debugger connects to a target, where that target might be actual target hardware or the software simulator. The target can be locally connected to the host PC, or remotely connected. That is, when it is remotely connected, it is running on a separate machine but accessible through the TCP/IP network by the host machine. In either case, the external tool is software that listens on a TCP/IP port for a connection from the debugger, and then relays data between the debugger and the target.

To create the external tool configuration for the Symphony SoundBite, perform the following steps.

1. Select the dropdown menu ***Run** > **External Tools** > **External Tools…***.

2. Select the External Tool template OpenOCD GDB Server.

3. Double-click the selection or click on the ***New Launch Configuration*** button. The ***New Launch Configuration*** button looks like a sheet of paper with a yellow “+” on it.

4. Choose ***56371*** in the ***Device*** pull down menu and choose ***soundbite*** in the ***Dongle*** pull down menu.

5. Connect the Symphony SoundBite board to the host PC and allow Windows time to recognize and install the drivers.

6. After the hardware is ready, click the ***Run*** button to launch the External Tool.

   After launching, the console tab should display a single status line with no errors similar to the following:

   ```
   Info: openocd.c:82 main(): Open On-Chip Debugger ps001 (2007-10-19 18:00 CEST)
   ```

If an error message occurs (displayed on the line below the info line above), re-launch the External Tool for Symphony SoundBite. Sometimes OpenOCD is unable to immediately connect to the debugger port of the DSP56371. Also, once an External Tool configuration has been launched, it can be re-launched by using the numbered entry in the ***Run** > **External Tools*** history submenu or the ***Run*** button in the button bar of the Symphony Studio window (the button that has a green circle with a white triangle in it with a little red toolbox in the lower right corner).

2.4 Debug Configuration

The ***Commands*** tab in Symphony Studio is where Gnu debugger commands can be specified to be executed at the beginning of a debug session (***Initialize commands***) or to be executed any time application code from a project is executed (***Run commands***). Since the debugger is not able to reset the DSP through the JTAG/OnCE port, the DSP56371 on the Symphony SoundBite requires the DSP to be forced into a reset-like state so that all queued interrupts are cleared and all the on-chip peripherals are reset. The processor state is not changed, so application code cannot assume that core registers are in their default reset states. The ***Initialize commands*** are used to force a reset-like state during the Initialize phase of debugging, before any application code is downloaded or run.

When using the **Freescale 56371** Debug configuration, Symphony Studio automatically configures the Debug launch configuration to force the DSP on the Symphony SoundBite into the reset-like state. You can verify this by performing the following steps:

1. Select the ***Run*** > ***Debug...*** menu to open the Debug configuration dialog box.

2. Select the Symphony SoundBite debug configuration. If this is not a selection, create it using the **Freescale 56371** template.

3. Click on the ***Commands*** tab. In the ***'Initialize' commands*** text box, verify that the following four lines of text are included:

   ```
   M p:0 0x000084
   M p:1 0x000200
   set $pc=0
   cont
   ```

   Line 1 writes the value 0x84, which is the opcode for the RESET instruction, at P:0.
   Line 2 writes the value 0x200, which is the opcode for the DEBUG instruction, at P:1.
   Line 3 changes the DSP's PC to 0.
   Line 4 instructs the debugger to tell the DSP to begin executing instructions normally, stopping at the DEBUG instruction at P:1.

   These four instructions are executed every time a Debug configuration is launched and before the project's application code is downloaded.